

How To Install Wordpress with nginx on CentOS 6

Authored by: **ASPHostServer Administrator** [asphostserver@gmail.com]

Saved From: <http://faq.asphosthelpdesk.com/article.php?id=290>

Wordpress is a free and open source website and blogging tool that uses php and MySQL. It was created in 2003 and has since then expanded to manage 22% of all the new websites created and has over 20,000 plugins to customize its functionality.

Setup

The steps in this tutorial require the user to have root privileges on your server.

Before working with wordpress, you need to have LEMP installed on your server. If you don't have the Linux, nginx, MySQL, PHP stack on your server

Once you have the user and required software, you can start installing wordpress!

Step One—Download WordPress

We can download Wordpress straight from their website:

```
wget http://wordpress.org/latest.tar.gz
```

This command will download the zipped wordpress package straight to your user's home directory. You can unzip it the the next line:

```
tar -xzf latest.tar.gz
```

Step Two—Create the WordPress Database and User

After we unzip the wordpress files, they will be in a directory called wordpress in the home directory.

Now we need to switch gears for a moment and create a new MySQL directory for wordpress.

Go ahead and log into the MySQL Shell:

```
mysql -u root -p
```

Login using your MySQL root password, and then we need to create a wordpress database, a user in that database, and give that user a new password. Keep in mind that all MySQL commands must end with semi-colon.

First, let's make the database (I'm calling mine wordpress for simplicity's sake; feel free to give it whatever name you choose):

```
CREATE DATABASE wordpress;  
Query OK, 1 row affected (0.00 sec)
```

Then we need to create the new user. You can replace the database, name, and password, with whatever you prefer:

```
CREATE USER wordpressuser@localhost;  
Query OK, 0 rows affected (0.00 sec)
```

Set the password for your new user:

```
SET PASSWORD FOR wordpressuser@localhost= PASSWORD("password");  
Query OK, 0 rows affected (0.00 sec)
```

Finish up by granting all privileges to the new user. Without this command, the wordpress installer will not be able to start up:

```
GRANT ALL PRIVILEGES ON wordpress.* TO wordpressuser@localhost IDENTIFIED BY  
'password';  
Query OK, 0 rows affected (0.00 sec)
```

Then refresh MySQL:

```
FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.00 sec)
```

Exit out of the MySQL shell:

```
exit
```

Step Three—Setup the WordPress Configuration

The first step to is to copy the sample WordPress configuration file, located in the WordPress directory, into a new file which we will edit, creating a new usable WordPress config:

```
cp ~/wordpress/wp-config-sample.php ~/wordpress/wp-config.php
```

Then open the wordpress config:

```
sudo nano ~/wordpress/wp-config.php
```

Find the section that contains the field below and substitute in the correct name for your database, username, and password:

```
// ** MySQL settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define('DB_NAME', 'wordpress');  
/** MySQL database username */  
define('DB_USER', 'wordpressuser');  
/** MySQL database password */  
define('DB_PASSWORD', 'password');
```

Save and Exit.

Step Four—Copy the Files

We are almost done uploading Wordpress to the server. We need to create the directory where we will keep the wordpress files:

```
sudo mkdir -p /var/www/wordpress
```

The final move that remains is to transfer the unzipped WordPress files to the website's root directory.

```
sudo cp -r ~/wordpress/* /var/www/wordpress
```

We can modify the permissions of `/var/www` to allow future automatic updating of Wordpress plugins and file editing with SFTP. If these steps aren't taken, you may get a "To perform the requested action, connection information is required" error message when attempting either task.

First, switch in to the web directory:

```
cd /var/www/
```

Give ownership of the directory to the nginx user, replacing the "username" with the name of your server user.

```
sudo chown nginx:nginx * -R  
sudo usermod -a -G nginx username
```

Step Five—Set Up Nginx Server Blocks

Now we need to set up the WordPress virtual host. Although Wordpress has an extra step in its installation, the nginx website gives us an easy configuration file: Open up the default nginx default hosts file:

```
sudo vi /etc/nginx/conf.d/default.conf
```

The configuration should include the changes below (the details of the changes are under the config information):

```
#  
# The default server
```

```
#
server {
    listen      80;
    server_name _;
    #charset koi8-r;
    #access_log logs/host.access.log main;
    location / {
        root    /var/www/wordpress;
        index index.php index.html index.htm;
    }
    error_page 404              /404.html;
    location = /404.html {
        root    /usr/share/nginx/html;
    }
    # redirect server error pages to the static page /50x.html
    #
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root    /usr/share/nginx/html;
    }
    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ /\.php$ {
    #    proxy_pass http://127.0.0.1;
    #}
    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #
    location ~ /\.php$ {
        root           /var/www/wordpress;
        fastcgi_pass   127.0.0.1:9000;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include        fastcgi_params;
    }
    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny  all;
    #}
}
```

- Here are the details of the changes—you may have some of these in effect already:
- Add index.php within the index line.
- Change the root to /var/www/wordpress;
- Uncomment the section beginning with "location ~ /\.php\$ {"
- Change the root to access the actual document root, /var/www/wordpress;

- Change the fastcgi_param line to help the PHP interpreter find the PHP script that we stored in the document root home.

Save, exit, and restart nginx for the changes to take effect:

```
sudo service nginx restart
```

Step Six—RESULTS: Access the WordPress Installation

Once that is all done, the wordpress online installation page is up and waiting for you:

Access the page by visiting your site's domain or your Virtual Private Server's IP address (eg. example.com) and fill out the short online form (it should look like [this](#)).